Teaching Distributed Data Mining on DAS: How to do it right?

Wojtek Kowalczyk
wojtek@liacs.nl

Universiteit Leiden

DAS Workshop, 13.02.2013, TU Delft

Why Distributed Data Mining?

- New Technologies : Internet, Telecom, Bioinformatics, RFIDs, Sensors, … => **new problems => new algorithms** (web search, text mining, recommender systems, social networks, DNA-sequences, microarrays, traffic data, …)

- Exponential growth of the gap between available data and data processing capabilities => **need for faster algorithms**

- Computers are changing: multi-cores, clusters, grids, clouds, … **=> another way of thinking about algorithms**

Exponential “data flood”

- Moore’s Law: “processing speed doubles every 18 months”
- Kryder’s Law: “storage capacity doubles every 12 months”
- Lyman & Varian (Berkeley, Google):
  “the amount of collected data doubles every 12 months”

- Therefore every 3 years processing speed increases 4 times, while the amount of available data increases 8 times

  => the **“data flood” is doubling every 3 years!!!**

Why DAS? It's obvious!

- A big, powerful, distributed machine
- Easily reachable by students (and staff)
- Available 24h/day
- For free …
- An "elite community": "he works on DAS!"…
- What else would be an alternative?

Teaching Distributed Data Mining

- Theory first: Advances in Data Mining (6 ECTS)
  - A regular course on mining big data
  - 12 lectures
  - 3 programming assignments (not on DAS!)
  - Written Exam
  - Final grade = 60% Assignments + 40%Exam

- Then practice: Seminar “Distributed Data Mining” (6 ECTS)
  - Team projects (on DAS4); 3-5 persons per team
  - Weekly progress meetings with presentations and discussions
  - Final Reports, Presentations, Software

Advances in Data Mining

- Introduction (1 lecture)
- Finding Similar Items (2 lectures); **First Assignment**
- Mining Data Streams (1 lecture);
- Recommender Systems (2 lectures); **Second Assignment**
- Mining Social Networks (3 lectures); **Third Assignment**
- Advertising on the Web (1 lecture)
- Distributed Data Mining: Hadoop, Map Reduce, GraphLab,…

Textbook:

Mining of Massive Datasets (Rajaraman, Leskovec, Ullman)

http://infolab.stanford.edu/~ullman/mmds.html

Seminar Distributed Data Mining (2012)

- Two teams (2x4 students)
- Two projects:
  - Netflix :
    reproduce a solution of the Netflix Challenge
    (a $1.000.000 data mining competition)
  - Wikipedia:
    very fast (milliseconds) detection of plagiarism using Wikipedia as a reference collection of documents

Netflix Challenge

- Netflix is the biggest DVD/movie rental company
- 20 million subscribers
- receive 10 million ratings a day
- generate 5 billion predictions per day **(DATA MINING!)**
- Accuracy of predictions and speed of the system is crucial for maintaining the competitive advantage!

  In 2006 Netflix Announced a $1.000.000 CHALLENGE:

  improve the Neflix Recommendation Engine by at least 10%
- The prize awarded in 2009 to a “team of teams”

Netflix Challenge on DAS-4

- Original Netflix data (100.000.000 records)
- Lots of papers, including detailed descriptions of the winning submission
- Several collections of recommender algorithms:
  - MyMediaLite (C#)
  - GraphLab (C)
  - Mahout (Java + MapReduce)
- Approach: train a few hundred models and blend them into a final solution (the most successful approach)
- That would cost a few thousand of CPU-time ...

Netflix Challenge:

- The software didn't work !!!
  - Too slow (weeks to build a simple model)
  - Bugs
  - Accuracy different than reported in papers!?

- Change of strategy: implement some of the most promising algorithms from scratch
  - Distributed Restricted Boltzmann Machines
  - Distributed Biased Matrix Factorization
  - Blending Strategies

Netflix Challenge: what we've learned?

- It was very fortunate that the software didn't work!
  - It forced us to study some algorithms in depth
  - When implementing our own algorithms we were forced to think very carefully about the speed and parallelism
  - We escaped boredom …
- When proposing a project think about:
  - - what will the students learn?
  - - will they enjoy it?
  - - is it challenging enough?

Plagiarism and Wikipedia

- Wikipedia: 4 million documents (38 GB)
- Use it as a reference corpora for detecting plagiarism
- Gain some hands-on experience with LSH (Locality Sensitive Hashing)

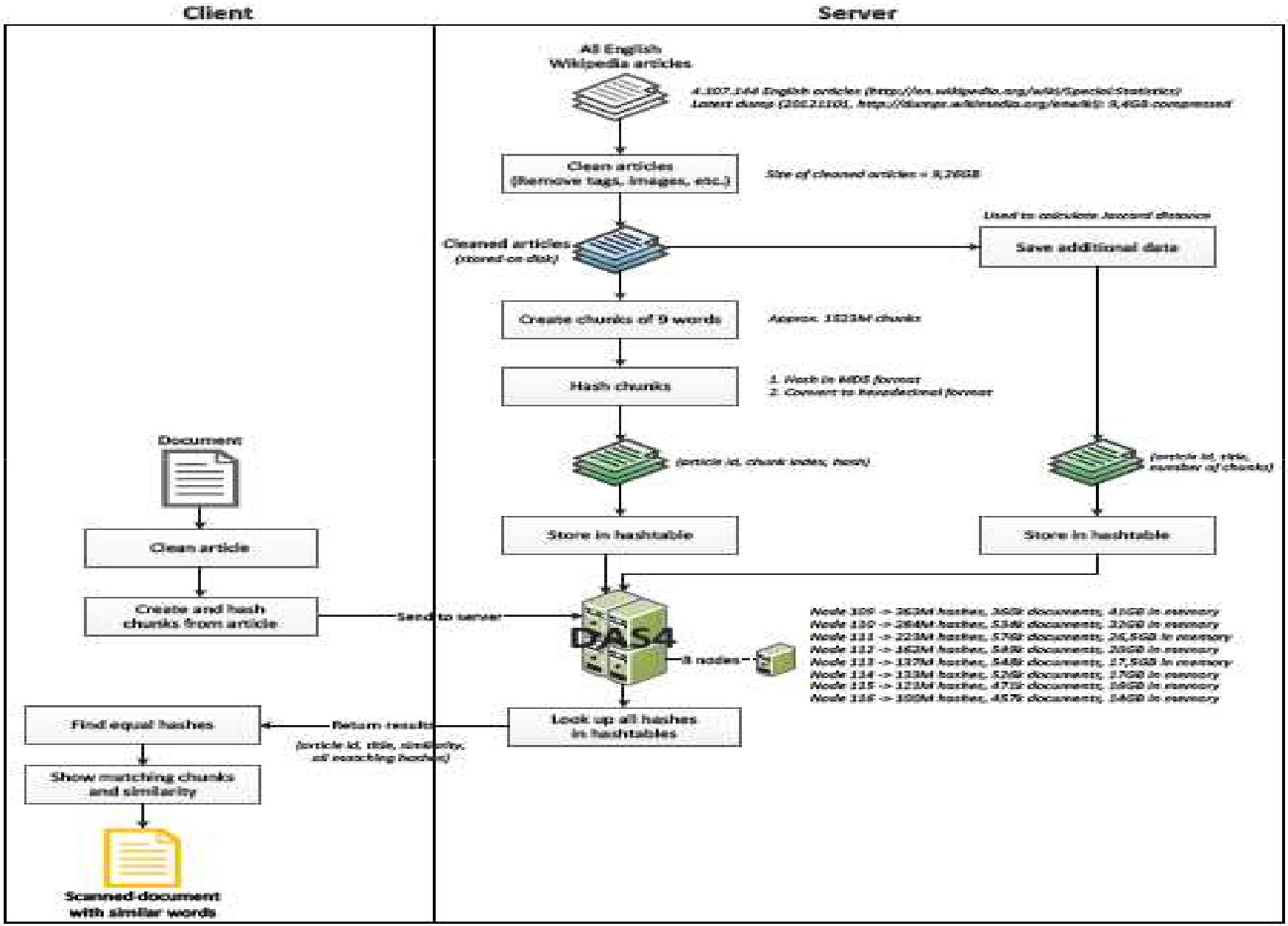
- LSH: hash documents in such a way that “similar” or “overlapping” documents fall into the same buckets

- Lookup time: O(1) (and O(N) memory)

Plagiarism and Wikipedia

- Implemented from scratch
  - MapReduce + Python: parsing and preprocessing
  - C/C++: distributed hash tables
- 38GB XML -> 9GB TXT -> 60GB shingles (9-words)
- LSH -> 200GB hashes to be kept in RAM (8 nodes)
- The distributed hash tables accessed in a client-server fashion

- Matching an input document against all 4.000.000 of Wikipedia documents takes about 1 millisecond


Client
Server
All English Wikipedia articles
4.107.144 English articles (https://en.wikipedia.org/wiki/Special:Statistics)
Latest dump (20221101, https://dumps.wikimedia.org/enwiki): 8,4GB compressed
Clean articles (Remove tags, images, etc.)
Size of cleaned articles = 8,2GB
Cleaned articles (stored on disk)
Used to calculate Jaccard distance
Save additional data
Create chunks of 9 words
Approx. 1023M chunks
Hash chunks
1. Hash in MD5 format
2. Convert to hexadecimal format
(article id, chunk index, hash)
(article id, title, number of chunks)
Store in hashtable
Store in hashtable
Document
Clean article
Create and hash chunks from article
Send to server
DAS4
8 nodes
Node 109 -> 262M hashes, 360K documents, 41GB in memory
Node 110 -> 284M hashes, 534K documents, 32GB in memory
Node 111 -> 223M hashes, 576K documents, 26,5GB in memory
Node 112 -> 162M hashes, 548K documents, 20GB in memory
Node 113 -> 137M hashes, 548K documents, 17,5GB in memory
Node 114 -> 133M hashes, 526K documents, 17GB in memory
Node 115 -> 123M hashes, 471K documents, 16GB in memory
Node 116 -> 102M hashes, 457K documents, 14GB in memory
Look up all hashes in hashtables
Return results
(article id, title, similarity, all matching hashes)
Find equal hashes
Show matching chunks and similarity
Scanned document with similar words

Wikipedia: what have we learned?

- A perfect project:
  - A non-trivial dataset
  - Learning a lot of new stuff: LSH, MapReduce, Python, DAS scheduler, coordinating DAS nodes, …
  - 1 millisecond plagiarism detection looks like a magic!
    (you compare one document with 4.000.000 others, actually just by calculating some hash functions and doing simple calculations)
  - Could serve as a starting point for experimenting with LSH applied to images, fingerprints, sounds, video, etc.

  - It's a pity that you can't use DAS interactively
    (a not via the DAS scheduler)...

How to do it right?

- Organize students in small groups (3-5)
- Select projects that are around the same theme (to increase inter-group synergy)
- Weekly meetings, presentations and discussions
- Project should be challenging (not “reproduce something”!), if possible exploring some “magic” algorithms or approaches
- Force students to produce a well documented software, reports, demo’s – they should be proud of their work!

Final observations

- DAS is a very attractive “educational tool”:
  - students learn much more than during a 'regular’ course
  - they broaden their horizons: "web-scale computing”
  - they are getting prepared for emerging market of the “big data mining”
  - they are getting prepared for academic research

- Universities should put more stress on training students in distributed computing (and distributed data mining) and make distributed platforms even more accessible…

How to do it right?

?